LOG4J - OVERVIEW

Log4j is a Reliable, Fast and Flexible Logging Framework (APIs) written in Java which is distributed under the Apache Software License.

Log4j has been ported to the C, C++, C#, Perl, Python, Ruby, and Eiffel languages.

Log4j is highly configurable through external configuration files at runtime. It views the logging process in terms of levels of priorities and offers mechanisms to direct logging information to a great variety of destinations, such as a database, file, console, UNIX Syslog etc.

Log4j has three main components:

- **loggers:** Responsible for capturing logging information.
- **appenders :** Responsible for publishing logging information to various preferred destinations.
- **layouts:** Responsible to format logging information in different styles.

History of log4j

- Started in early 1996 as tracing API for the E.U. SEMPER (Secure Electronic Marketplace for Europe) project.
- After countless enhancements, several incarnations and much work that API has evolved to become log4j, a popular logging package for Java.
- The package is distributed under the Apache Software License, a fully-fledged open source license certified by the open source initiative.
- The latest log4j version, including full-source code, class files and documentation can be found at http://logging.apache.org/log4j/.

log4j Features:

- log4j is thread-safe.
- log4j is optimized for speed.
- log4j is based on a named logger hierarchy.
- log4j supports multiple output appenders per logger.
- log4j supports internationalization.
- log4j is not restricted to a predefined set of facilities.
- Logging behavior can be set at runtime using a configuration file.
- log4j is designed to handle Java Exceptions from the start.
- log4j uses multiple levels, namely ALL, TRACE, DEBUG, INFO, WARN, ERROR and FATAL.
- The format of the log output can be easily changed by extending the *Layout* class.
- The target of the log output as well as the writing strategy can be altered by implementations of the Appender

interface.

- log4j is fail-stop. However, altough it certainly strives to ensure delivery, log4j does not guarantee that each log statement will be delivered to its destination.

Pros N Cons of Logging:

Logging is an important component of the software development. A well written logging code offers quick debugging, easy maintenance, and structured storage of an application's runtime information.

Logging does have its drawbacks also. It can slow down an application. If too verbose, it can cause scrolling blindness. To alleviate these concerns, log4j is designed to be reliable, fast and extensible.

Since logging is rarely the main focus of an application, the log4j API strives to be simple to understand and to use.